融合 Word2vec 与 TextRank 的关键词抽取研究

宁建飞　刘降珍
(罗定职业技术学院电子信息系　罗定 527200)

摘要:【目的】通过融合单个文档内部结构信息和文档整体的词向量关系进行关键词抽取。【方法】利用 Word2vec 将文档集中所有词汇进行向量表征,并且通过词向量计算词汇之间的相似度,进而对 TextRank 算法进行改进,将候选关键词的权重按照词汇之间的相似度和邻接关系进行非均匀分配,并构建对应的概率转移矩阵用于词汇图模型的迭代计算以及关键词抽取。【结果】实现 Word2vec 与 TextRank 的有效融合,且当训练文档集词汇分布合理时,关键词抽取效果较明显。【局限】需要进行成本较高的文档集训练,获取词向量以及词关系矩阵。【结论】文档集中的词关系有助于修正单文档内部的词关系,提升单文档的关键词抽取准确性。

关键词:关键词抽取　Word2vec　TextRank　图模型　词向量
分类号:TP391　G250

1 引　言

抽取关键词的目的在于高度凝练文本的主题,快速获取文本的核心内容。关键词抽取在新闻、学术论文的自动摘要,社会化标签标注,文本主题抽取等领域具有重要作用。

常见的关键词抽取步骤为:对文本进行分词,去除无用的停用词,判断词是否为关键词,选择 N 个词作为该文本关键词。其中,判断所分的词是否为关键词,可以通过关键词标记语料进行分类模型训练,通过分类模型进行判断;也可以通过结合文本内部词与词之间的关系,以图模型的方式进行识别。而图模型的实现方法又以 TextRank[1]为典型代表。

经典的 TextRank 算法不依赖于其他训练语料,重点研究文本内部词语结构关系,建立图模型进行关键词抽取。夏天[2]的研究成果指明词语本身的重要差异会影响相邻节点的影响力传递,顾益军等[3]将 TextRank 算法与 LDA 相结合,将候选词语节点的重要性按照文档集主题分布进行非均匀转移。

为了能够充分研究词汇与词汇之间的关系,借助文档本身以及文档集所提供的外部信息,本文将 Word2vec[4]与 TextRank 算法进行融合,通过 Word2vec 对外部文档集进行词向量表征,获取词汇之间的相似度,对 TextRank 算法进行改进,将候选词汇节点的权重按照相邻词的相似度进行合理分配,通过迭代计算每个词语权重,最终通过权重重新排序,获取关键词抽取结果。

2 研究背景

文本关键词抽取从语料是否被标记的角度可分为有监督和无监督两种。其中有监督的关键词抽取典型代表可以把关键词抽取看作是一个二分类问题[5-6],对于任何一个文本中的词汇,进行二值判断,即属于关键词还是非关键词二值分类,这种方法要求对文档集语料提前进行关键词人工标记,进行分类模型训练,进而实现关键词抽取,需要大量的人工干预,代价较高。

在无监督的关键词抽取领域,国内已经有不少相

通讯作者:宁建飞,ORCID:0000-0001-9941-3670,E-mail:ningafei@126.com。

关研究。耿焕同等[7]利用词共现图形成的主题信息与主题间连接关系，自动提取文档的主题词。刘菲等[8]提出利用关联规则挖掘算法进行主题词提取。蒋昌金等[9]考虑词语的语义信息，提出一种基于组合词和同义词的主题词提取算法。

目前比较主流的无监督关键词抽取基础方法主要有三种：基于词频统计的 TF-IDF 模型关键词抽取[10]、基于主题模型的关键词抽取和基于词汇图模型的关键词抽取。在三种主流的无监督关键词抽取研究之上，又有很多其他相关的优化算法。

基于词频统计的 TF-IDF 模型关键词抽取是一种简单而又经典的关键词抽取方法，通过词频提升重要词汇的权重，通过逆向文档频降低公共词的权重，但这种方法基于词频，对于短文本效果并不好，且其忽略了文本内部词汇与词汇之间的关系。

基于 LDA[11]隐含主题模型的关键词抽取逐渐受到人们的重视[12-13]，LDA 的主题模型通过语料训练得到，获取"文档–主题"概率矩阵以及"主题–词汇"概率矩阵，进而求得"文档–词汇"概率和矩阵，并进行关键词抽取，关键词抽取的效果与训练文档集的主题分布强相关。

基于词汇图模型关键词抽取不需要额外的文档集进行训练，只依靠自身文本词汇结构信息即可进行关键词抽取，简单而有效，所以得到广泛的应用，其中又以 TextRank[1]算法为典型代表。

随着深度学习的兴起，刘俊等[13]使用深度学习工具 Word2vec 进行关键词抽取，使用 Word2vec 将训练文档集中所有词汇进行 K 维向量表征，基于词向量进行词汇之间的相似度计算，进而实现词汇聚类得到文档的关键词。

文献[2]在 TextRank 的基础上，提出词汇本身的重要性差异会影响相邻节点的影响力传递结果。文献[12]利用 LDA 主题模型进行关键词抽取，需要依赖于大量的训练数据，代价较高，且无法满足单文档的关键词抽取需求。而文献[3]在保持 PageRank 的均匀跳转的假设下，采用LDA 隐含主题模型分析计算词汇的整体影响力，结合词语之间的邻接关系改进 TextRank 的概率转移矩阵，但这种方法没有考虑词汇与词汇之间的整体分布关系。文献[14]使用深度学习结合词汇聚类的方法进行关键词抽取，研究表明对于篇幅较长的文章效果较好，但对于篇幅较短的文章则无法满足关键词准确抽取的需求。

基于文献[3]的基本研究思路，借助单一文档的内部结构信息和文档整体的信息进行主题词抽取。本文提出词汇节点关系受文档集词汇之间关系分布影响，结合 Word2vec[4,15]训练得到的词汇相似度矩阵，改进 TextRank 词汇节点的初始权重以及概率转移矩阵，同时考虑单文档内部词汇结构以及文档集词汇结构信息，进而提升关键词抽取效果。

3 研究框架与方法

本文借鉴顾益军等[3]的研究思路，融合单一文档的内部结构信息与文档的整体信息，进行主题词抽取。研究文档集合词汇节点之间的关系分布对单一文档词汇结构的影响。

TextRank 算法的核心思想来源自著名的网页排名算法 PageRank[16]。TextRank 算法将文本拆分成最小组成单元，即词汇，作为网络节点，组成词汇网络图模型。TextRank 在迭代计算词汇权重时与 PageRank 一样，理论上是需要计算边权的，但是为了简化计算，通常会默认相同的初始权重，以及在分配相邻词汇权重时进行均分。

本文使用 Word2vec 算法进行文档集词向量计算，获取文档集词汇之间的相似矩阵，用于改进 TextRank 算法的初始权重计算以及迭代计算的概率转移矩阵，最终获取到文档内部所有有效词汇的权重，进行关键词抽取。

Word2vec 是 Google 在 2013 年开源的一款将词表征为空间向量的模型工具，主要采用连续词袋模型[17] (Continuous Bag-Of-Words, CBOW)以及 Skip- gram[17]模型。它是一种深度学习的模型，基于人工神经网络，通过多层感知机将初始的底层特征组合为更抽象的高层特征，并将高层特征用于普通的机器学习方法以得到更好的效果[14]。Word2vec 通过训练，可以把文本内容的处理简化为 K 维向量空间中的向量运算，向量空间上的相似度可以用来表示文本语义上的相似度。

CBOW 模型的目的是通过上下文预测当前词汇出现的概率。从图 1 可以看到 CBOW 模型的网络结构包括三层：Input Layer(输入层)；Projection Layer(投影

层); Output Layer(输出层)。其中，训练样本为(Context(w), w)，假设: Context(w)是由 w 前后各 c 个词构建而成。

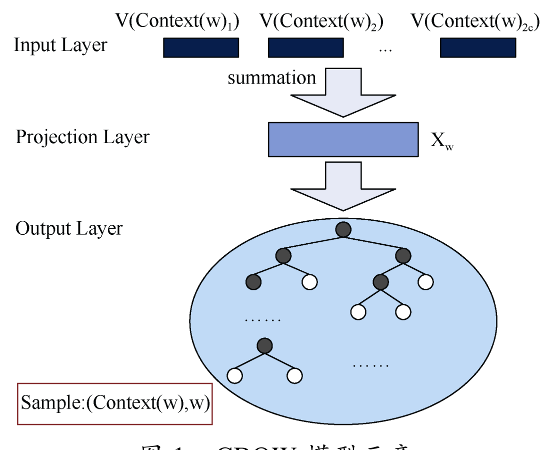


图 1　CBOW 模型示意

(1) 输入层: 输入层包括样本 Context(w)中 2c 个词的词向量，即图 1 中 Input Layer 层所示 $V(Context(w)_1)$, $V(Context(w)_2)$, …, $V(Context(w)_{2c}) \in R^m$。其中, m 代表词向量的长度。

(2) 投影层: 在投影层进行的操作是将 Input Layer 阶段的 2c 个词向量做求和操作，如公式(1)所示:

$$X_w = \sum_{i=1}^{2c} V(Context(w)_i) \in R^m \qquad (1)$$

其中，X_w 为词 w 向量累加和，向量总数为 2c 个，$V(Context(w)_i)$是样本文档的词向量表示，m 为向量长度, R 为词典向量范围。

(3) 输出层: 输出层表示的是一棵二叉树，它以训练样本中出现的词作为叶子节点，以各词在语料中出现的次数作为权值进行 Huffman 树构造。在这棵 Huffman 树中，叶子节点共有 N(=|D|)个，分别对应词典 D 中的词，非叶子节点共有 N–1 个，即图 1 中标成黑色的节点。

Skip-gram 模型的提出是为了解决训练语料选择的问题。笔者在选择 Word2vec 的训练文档集时，要求做到语料文档集覆盖度必须高、语料文档集必须足够准确。在 N 元模型中，固定窗口大小的局限在于窗口范围外的词汇关系不能正确地被反映到模型中，可以通过增加词汇窗口大小来降低影响，但单纯增加窗口 N 值会提升训练的复杂度。而 Skip-gram 模型的提出很好地解决了这一问题。

由图 2 可以看到, Skip-gram 同样由三层网络模型构成，包括: 输入层、投影层、输出层。其中输入层(Input Layer)的输入是词向量 $W(t) \in R^m$，投影层的训练目的是使公式(2)的值最大。

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{-c \leqslant j \leqslant c} \log p(W_{t+j} \mid W_t) \qquad (2)$$

其中，c 是词汇窗口大小，在 Skip-gram 模型中即指 n-Skip-gram 的 n 值大小，T 是训练文档集的大小。Skip-gram 模型中计算词汇条件概率如公式(3)所示:

$$p(w_O \mid w_I) = \frac{v_{w_O}^{`T} v_{w_I}}{\sum_{w=1}^{|v|} \exp(v_{w_O}^{`T} v_{w_I})} \qquad (3)$$

其中，v_w 和 $v_w^`$ 分别是词 w 的输入和输出向量。与 CBOW 模型一样，Skip-gram 模型的输出层(Output Layer)也是一棵 Huffman 树。

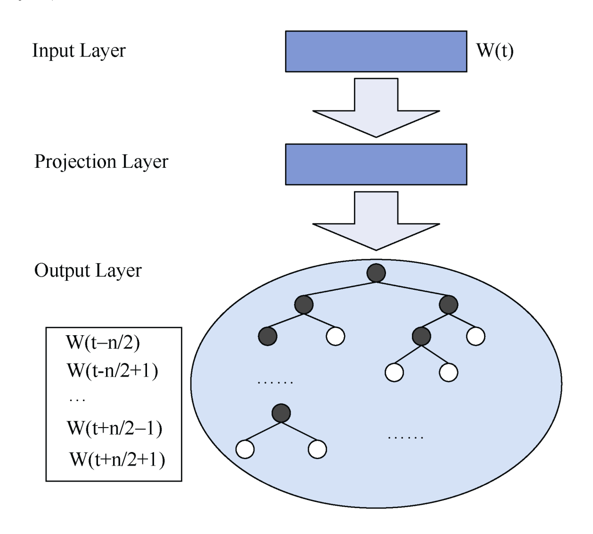


图 2　Skip-gram 模型示意

4　研究过程

根据文献[2-3,10]，将关键词抽取的问题转换为文档词汇重要性排序问题，根据权重对词汇进行排序，获取 TopN 个词作为文档的关键词。基于文献[1]，构建一个以词汇作为网络节点的关键词图，通过迭代计算获取每个词汇的权重。将 Word2vec 训练出来的词汇相似度矩阵融合到迭代计算中，优化权重计算结果。首先，需要构建一个关键词图，在构建关键词图之前，对训练文档集以及测试文档集进行预处理，预处理过程分为以下 4 个步骤:

(1) 通过中国科学院计算技术研究所 ICTCLAS[①]分词工具对N篇文档组成训练文档集以及单篇测试文档进行分词，并使用停用词表过滤分词结果中的停用词，获得词汇集S_1与S_2，其中S_1由N个子词汇集组成，一个子词汇集对应一篇训练文档；

(2) 对词汇集 S_1 与 S_2 进行词性标注，保留重要词汇，如名词、动词、形容词，获得词汇集$S_1^{`}$与$S_2^{`}$，同上，$S_1^{`}$由 N 个子词汇集组成；

(3) 对词汇集 $S_1^{`}$ 与 $S_2^{`}$ 进行词汇去重，获得词典 $D=[w_1,w_2,\cdots w_m]\in(S_1^{`}\cup S_1^{`})$，即候选关键词；

(4) 使用 Word2vec 对 $S_1^{`}$ 进行训练，得到词向量，进而得到词典 D 的词汇相似度矩阵。

通过 CBOW 模型以及 Skip-gram 模型进行样本文档集训练，对词典 D 中的每一个词进行 K 维词向量表征，然后通过计算余弦夹角，得到词典 D 中每个词与其他词汇之间的相似度，如公式(4)所示：

$$\mathrm{Sim}(e_i,f_j)=\cos\theta=\frac{e_i\cdot f_j}{\|e_i\|\cdot\|f_j\|} \tag{4}$$

其中，e_i是源文档句子中第 i 个词，f_j是目标文档句子中第 j 个词，第 i 个词与第 j 个词之间的相似度为 $\mathrm{Sim}(e_i, f_j)$，而 e_i, f_j 为词向量表示。

假设词典总大小为 n，则通过 Word2vec 的文档集训练，获得一个 n×n 的词汇相似度矩阵，通过矩阵可以得到词典中任意两个词汇之间相似度，如公式(5)所示：

$$M(\mathrm{Sim}(w_iw_j))=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \tag{5}$$

其中，$M(\mathrm{Sim}(w_i\ w_j))$表示词典的相似度矩阵，$w_{ij}$表示词 ij 的相似度。需要注意的是，在矩阵中，下标相同的值表示同一词汇与自身的相似度，例如 w_{ii} 表示词 i 与自身的相似度值，通常表示为 1，无参考意义，可忽略。

在所有预处理工作完成之后，进行测试文档候选关键词图构建。TextRank 的核心思想是一个词汇节点的重要性取决于有多少个相邻节点指向该节点，且相邻节点的权重同样影响该节点，而词汇节点的权重计算如公式(6)所示：

$$R(w_i)=\gamma\sum_{j:w_j\to w_i}\frac{e(w_j,w_i)}{O(w_j)}R(w_j)+(1-\gamma)\frac{1}{|V|} \tag{6}$$

其中，$R(w_i)$是词 w_i 的权重，$O(w_j)$为词 w_i 的出度，$e(w_j, w_i)$为 $w_j\to w_i$ 边权，V 为词汇节点集合，$\gamma\in[0,1]$为平滑因子，即阻尼系数(Damping Factor)，通常取值为 0.85。

传统的 TextRank 中，将每个词汇节点的权重默认为 1，通过相邻关系进行迭代计算，更新节点的权重，在计算词汇节点的权重贡献时以权重均分的形式向相邻节点传递。例如，图 3 为由 6 个词汇节点{V, V_1, V_2, V_3, V_4, V_5}组成的候选关键词图初始状态，初始默认每个词汇节点权重为 1，权值向相邻节点均分传递，所以 V 节点指向其他 5 个节点的边权设置为 0.2，而其他 5 个词汇节点指向 V 节点的边权为 1，后续迭代计算过程类似，同样以权重均分的形式设置指向相邻节点的边权。

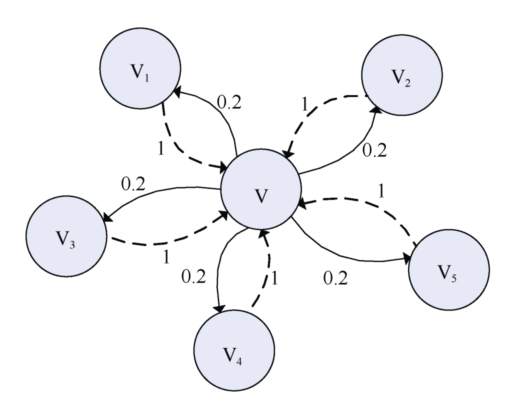


图 3　传统候选关键词图初始状态

本文基于当前的候选关键词图，讨论如何优化词汇节点的初始权重，以及优化词汇节点影响力传递方式，改进最终词汇权重排序效果，进行关键词抽取。对于关键词图词汇节点的状态初始化，更为合理的方式并不是默认为 1，而是把各个节点之间相互影响力作为初始状态，在这里词汇节点之间的相互影响力可以使用词汇之间的相似度进行量化，词汇节点初始权重计算如公式(7)所示。

①http://ictclas.nlpir.org/.

$$S(w_j)=\sum_{j:w_j\to w_i} e(w_j,w_i) \tag{7}$$

其中，$S(w_j)$为词汇节点 w_j 的初始权重，$e(w_j,w_i)$ 为 $w_j \to w_i$ 相似度。

在改进的转移矩阵中，引入词汇相似度进行迭代计算，词汇节点的权值分配受两个因素影响：一部分是词汇节点本身的重要性，代表文档内部结构的影响力，通常通过相邻节点进行调整，初始状态值可通过公式(7)计算得到，后续通过迭代计算获取得到，记为 $TP(w_i)$；另一部分则是词汇之间硬性关系影响力分值，可以通过 Word2vec 训练得到，形式如公式(5)所示，代表外部文档对词汇之间关系的影响，记为 $M(Sim(w_iw_j))$。因此，重新定义节点重要性迭代计算的过程如公式(8)所示：

$$TP(w_i)=\gamma\left(\alpha\sum_{j:w_j\to w_i}\frac{M(Sim(w_iw_j))}{O(M(Sim(w_iw_j)))}TP(w_j)+\beta\sum_{j:w_j\to w_i}\frac{1}{O(w_j)}R(w_j)\right)+(1-\gamma)\frac{1}{|V|} \tag{8}$$

其中，$\gamma\in[0,1]$ 为平滑因子，α和β是两种影响因素的权重因子，这里使$\alpha+\beta=1$，在实验中各取 0.5，即词汇节点本身影响与外部文档词汇关系影响各占 50%，$M(Sim(w_i\ w_j))$为外部文档词汇之间相似度，即词汇 w_j 与 w_i 的相似度，取值参考公式(5)，$O(w_j)$是 w_j 的出度，$R(w_j)$是词 w_j 的权重，V 为词汇节点集合。

在迭代计算之前，构建词汇之间的概率转移矩阵，如公式(9)所示：

$$M(T(w_iw_j))=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \cdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \tag{9}$$

其中，元素 w_{ij} 表示节点 w_j 的影响力转移到第 i 个词汇 w_i 的概率，具体体现为相邻词汇边权的分配，可以通过公式(10)计算得到。

$$w_{ij}=\alpha\frac{M(Sim(w_iw_j))}{O(M(Sim(w_iw_j)))}+\beta\frac{1}{O(w_j)} \tag{10}$$

在引入转移概率矩阵之后，每一次的迭代结果都可以通过 $M(T(w_iw_j))$计算得到，令 B_i 表示一次迭代的结果，则迭代计算的过程由公式(6)转换为公式(11)。

$$B_i=\gamma M(T(w_i\ w_j))\times B_{i-1}+(1-\gamma)\frac{e}{k} \tag{11}$$

其中，e 为一个所有分量为 1，维数为 k 的向量。当迭代计算相邻两次的计算结果差异较小时，停止迭代计算，即迭代计算结果已经收敛。在收敛之后，对所有词汇节点的当前权重进行降序排列，选取 TopN 个词作为文档的关键词进行输出。

整个 Word2vec 与 TextRank 的融合过程分两步：基于 Word2vec 对训练文档集进行训练，最终获取形如公式(5)的词汇关系矩阵；把外部词汇关系影响力带入公式(10)，通过迭代计算实现词汇节点的权重计算，进而进行权重排序以及进行关键词抽取。

5 实验结果

选取网络公开搜狗实验室语料集作为训练文档集以及测试文档集，范围覆盖军事、教育、经济、娱乐等多个领域。对于每个领域语料集，挑选 10 篇文档作为测试集，共 90 篇测试文档，其余作为训练文档集，共 4 500 篇文档。在 4GB 内存的计算机上对训练文档集进行 Word2vec 词向量训练，历时 38 分钟产生一个大小约 120MB 的词相似矩阵模型文件。

通过词相似矩阵模型对 TextRank 计算过程优化，对于 90 篇测试集文档最终自动提取 3、5、7、10 个关键词。采用多组人工对测试集文档进行关键词标注的形式，进行结果交叉验证，以降低个人主观性带来的结果偏差，最终分别提取 3、5、7、10 个关键词作为测试校验对比结果。

此外基于相同的训练文档集及测试集，本文实现了基于 TF-IDF 的关键词抽取算法，传统的 TextRank 关键词抽取算法，以及基于 Word2vec 词聚类关键词抽取算法，并对这 4 种关键词抽取算法的输出结果进行分析比较。

目前关键词抽取算法效果的评判标准有准确率 P、召回率 R 以及 F 值，计算公式如下：

$$P=\frac{\text{抽取结果中与人工标注相同的关键词个数}}{\text{人工标注的关键词总个数}} \tag{12}$$

$$R=\frac{\text{抽取结果中与人工标注相同的关键词个数}}{\text{抽取关键词总个数}} \tag{13}$$

$$\text{F-measure}=\frac{2PR}{P+R} \tag{14}$$

为了保证关键词评价的正确性，通过多组实验人

员交叉对测试文档集进行关键词人工标注，并且分别为测试文档集每篇文档标注了 3、5、7、10 个关键词。最后分别使用 4 种算法进行关键词提取，计算准确率、召回率以及 F 值三个评价指标。

由表 1–表 3 可以看出，基于词频统计的 TF-IDF 算法随着关键词数的增加，效果逐渐变差，且整体效果较差；而基于 TextRank 算法的关键词抽取效果变化波动不大；基于 Word2vec 词聚类的关键词抽取效果随着关键词数的增加，抽取效果逐渐变好；基于 Word2vec 与 TextRank 算法融合的关键词抽取效果随着关键词数的增加，抽取效果逐渐变好，且整体效果较好。

表 1　4 种算法准确率比较

算法 \ 个数	3	5	7	10
TF-IDF	0.305	0.263	0.241	0.238
TextRank	0.332	0.329	0.323	0.321
Word2vec	0.275	0.303	0.321	0.357
Word2vec+TextRank	0.314	0.336	0.376	0.398

表 2　4 种算法召回率比较

算法 \ 个数	3	5	7	10
TF-IDF	0.312	0.272	0.248	0.231
TextRank	0.327	0.334	0.331	0.323
Word2vec	0.281	0.311	0.327	0.346
Word2vec+TextRank	0.312	0.339	0.383	0.395

表 3　4 种算法 F 值比较

算法 \ 个数	3	5	7	10
TF-IDF	0.308	0.268	0.244	0.234
TextRank	0.330	0.332	0.326	0.322
Word2vec	0.278	0.306	0.324	0.352
Word2vec+TextRank	0.312	0.338	0.380	0.396

针对这种现象，本文对关键词抽取过程进行了深入分析，并得出如下结论：

(1) 传统的基于词频统计的 TF-IDF 算法关键词抽取效果比较一般。

(2) 基于传统词图模型 TextRank 算法关键词抽取效果比较稳定。

(3) 基于词向量聚类的关键词抽取算法适用于篇幅较大的文档。

(4) 融合 Word2vec 词向量与 TextRank 图模型的关键词抽取方法，在继承了 TextRank 关键词抽取效果稳定的基础上，抽取效果有了进一步的提升，同样也继承了 Word2vec 词聚类关键词抽取随着关键词数上升效果有所上升的特点。

融合了 Word2vec 与 TextRank 算法的关键词抽取，利用 Word2vec 进行词向量训练，进而计算词汇之间的相似度矩阵，因此本文提出如下改进：

(1) 使用更大量的训练文档集训练更精确的相似度矩阵。

(2) 在词典筛选上使用更精确的停用词字典，进一步排除无效词的干扰。

(3) 改进 Word2vec 算法模型，增加神经网络的层次，提高词向量的语义抽象层次，进一步提高词汇相似度的准确率。

(4) 使用分布式架构，例如 Spark 内存计算进行算法实验，提高算法的运行速度。

综上所述，融合 Word2vec 词向量与 TextRank 图模型的关键词抽取方法主要优势在于结合文档内部结构与外部文档词汇关系的影响，继承了两种算法的优势，关键词抽取效果相对较好，但本文结果以及算法分析依然存在很大的改进之处。

6　结　语

文档的本身结构信息与外部文档集体现的词汇关系是关键词抽取的重要依据。本文基于 Word2vec 进行词典词汇之间关系计算，进而改进 TextRank 算法的权重分配迭代计算公式，把词汇之间的相似度影响力纳入词汇节点边权分配转移构建中，通过迭代计算致使词汇节点权重收敛，进行词汇节点权重排序和关键词抽取，进而在相同文档集上使用不同算法进行对比分析。

实验结果表明，在随着关键词数的上升，本文方法略优于传统 TextRank 词图模型和 Word2vec 词聚类方法，且该方法继承了传统 TextRank 算法和 Word2vec 词聚类算法的优点。同时，训练文档集的规模以及文档内部结构和外部文档词汇关系影响力权重比对抽取结果影响也较大，因此，进一步提升训练文档集规模以及研究文档内外部影响力比重对抽取效果的影响，将是本研究后续的工作之一。

参考文献:

[1] Mihalcea R, Tarau P. TextRank: Bringing Order into Texts [C]. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain. 2004: 404-411.

[2] 夏天. 词语位置加权 TextRank 的关键词抽取研究[J]. 现代图书情报技术, 2013(9): 30-34. (Xia Tian. Study on Keyword Extraction Using Word Position Weighted Text Rank [J]. New Technology of Library and Information Service, 2013(9): 30-34.)

[3] 顾益军, 夏天. 融合 LDA 与 TextRank 的关键词抽取研究[J]. 现代图书情报技术, 2014(7-8): 41-47. (Gu Yijun, Xia Tian. Study on Keyword Extraction with LDA and TextRank Combination [J]. New Technology of Library and Information Service, 2014(7-8): 41-47.)

[4] Goldberg Y, Levy O. Word2vec Explained: Deriving Mikolov et al. 's Negative-sampling Word-embedding Method [OL]. ArXiv, 2014. arXiv: 1402.3722v1.

[5] Frank E, Paynter G W, Witten I H, et al. Domain-Specific Keyphrase Extraction [C]. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden. San Francisco: Morgan Kaufmann Publishers Inc., 1999: 668-673.

[6] Turney P D. Learning Algorithms for Keyphrase Extraction [J]. Information Retrieval, 2000, 2(4): 303-336.

[7] 耿焕同, 蔡庆生, 于琨, 等. 一种基于词共现图的文档主题词自动抽取方法[J]. 南京大学学报: 自然科学版, 2006, 42(2): 156-162. (Geng Huantong, Cai Qingsheng, Yu Kun, et al. A Method Based on the Co-occurrence of Automatic Text Keyphrase Extraction Method [J]. Journal of Nanjing University: Natural Science Edition, 2006, 42(2): 156-162.)

[8] 刘菲, 黄萱菁, 吴立德. 利用关联规则挖掘文本主题词的方法[J]. 计算机工程, 2008, 34(7): 81-83. (Liu Fei, Huang Xuanjing, Wu Lide. The Method of Using Association Rule Mining Text Topic Words [J]. Computer Engineering, 2010, 27(8): 2853-2856.)

[9] 蒋昌金, 彭宏, 陈建超, 等. 基于组合词和同义词集的关键词提取算法[J]. 计算机应用研究, 2010, 27(8): 2853-2856. (Jiang Changjin, Peng Hong, Chen Jianchao, et al. Keyword Extraction Algorithm Based on Combination of Words and Synonyms [J]. Computer Application Research, 2010, 27(8): 2853-2856.)

[10] 徐文海, 温有奎. 一种基于 TFIDF 方法的中文关键词抽取算法[J]. 情报理论与实践, 2008, 31(2): 298-302. (Xu Wenhai, Wen Youkui. Chinese Keywords Extraction Based on TFIDF Method [J]. Information Studies: Theory & Application, 2008, 31(2): 298-302.)

[11] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet Allocation [J]. Journal of Machine Learning Research, 2003, 3: 993-1022.

[12] 石晶, 李万龙. 基于 LDA 模型的主题词抽取方法[J]. 计算机工程, 2010, 36(19): 81-83. (Shi Jing, Li Wanlong. Topic Words Extraction Method Based on LDA Model [J]. Computer Engineering, 2010, 36(19): 81-83.)

[13] 刘俊, 邹东升, 邢欣来, 等. 基于主题特征的关键词抽取[J]. 计算机应用研究, 2012, 29(11): 4224-4227. (Liu Jun, Zou Dongsheng, Xing Xinlai, et al. Keyphrase Extraction Based on Topic Feature [J]. Application Research of Computers, 2012, 29(11): 4224-4227.)

[14] 李跃鹏, 金翠, 及俊川. 基于 Word2vec 的关键词提取算法[J]. 科研信息化技术与应用, 2015(4): 54-59. (Li Yuepeng, Jin Cui, Ji Junchuan. A Keyword Extraction Algorithm Based on Word2vec [J]. E-science Technology & Application, 2015(4): 54-59.)

[15] 周练. Word2vec 的工作原理及应用探究[J]. 科技情报开发与经济, 2015(2): 145-148. (Zhou Lian. Exploration of the Working Principle and Application of Word2vec [J]. Sci-Tech Information Development & Economy, 2015(2): 145-148.)

[16] Page L, Brin S, Motwani R, et al. The PageRank Citation Ranking: Bringing Order to the Web [R]. Stanford InfoLab, 1999.

[17] Tomas M, Kai C, Greg C, et al. Efficient Estimation of Word Representations in Vector Space [OL]. ArXiv, 2013. arXiv: 1301.3781v3.

作者贡献声明:

宁建飞: 提出研究思路, 设计研究方案, 实验及数据分析, 起草、撰写论文;
刘降珍: 论文修订。

利益冲突声明:

所有作者声明不存在利益冲突关系。

支撑数据:

支撑数据见期刊网络版 http://www.infotech.ac.cn。

[1] 宁建飞. train_set.zip. 训练文档集–搜狗实验室提供的 4500 篇覆盖多个领域的文档.

[2] 宁建飞. validation_set.zip. 测试文档集–搜狗实验室提供的 90 篇覆盖多个领域的文档.

收稿日期: 2016-03-01
收修改稿日期: 2016-04-19

Using Word2vec with TextRank to Extract Keywords

Ning Jianfei Liu Jiangzhen
(Department of Electronic Information, Luoding Polytechnic, Luoding 527200, China)

Abstract: [**Objective**] This study extracts keywords through combining the internal structure of each single document and the word vector of the corpus. [**Methods**] First, we used Word2vec to represent all words' vector from the document corpus and then calculated their similarities. Second, modified the TextRank algorithm and assigned weights to the keywords in accordance with their similarities and adjacency relations. Finally, we built a probability transfer matrix for the iterative calculation of the lexical graph model and then extracted keywords. [**Results**] The Word2vec and TextRank were integrated and extracted keywords effectively. [**Limitations**] The proposed method needs much training with the corpus to establish word vector and relation matrix. [**Conclusions**] The relationship among words from the document sets could help us modify the words relationship from a single document, and then increase the accuracy of extracting keywords from the individual document.

Keywords: Keyword extraction Word2vec TextRank Graphical model Word vector

OCLC 和 RLUK 发布研究报告，分析英国研究图书馆馆藏情况

OCLC 和英国研究图书馆(RLUK)于近日发布了一份新的研究报告，该报告全面分析了英国研究图书馆的馆藏情况，在广度、深度和复制呈现方面有其独一无二的视角，并且突出强调了将这些馆藏资源作为一个共有资源所面临的机会和挑战。该报告题为《有力的数字：英国研究图书馆馆藏》(Strength in Numbers: The Research Libraries UK (RLUK) Collective Collection)。

该报告描述了 RLUK 集体馆藏，即 RLUK 成员图书馆的联合收藏的突出特征，并且特别强调了馆藏印刷资源的特征。报告中的发现将会为 RLUK 成员馆的战略决策提供支持，特别是在成员馆就长期馆藏管理(长期保存和存储)进行深度合作，数字馆藏替代印刷馆藏的可能性，如何更有效地利用图书馆空间等方面。

报告的调查结果包括：

(1) RLUK 集体馆藏包含 2 940 万份互不相同的出版物(各种类型都有)，其中包含 2 090 万份互不相同的印刷图书；

(2) RLUK 成员馆馆藏的印刷图书丰富多样，共有 467 种语言，出版自 254 个国家和地区；

(3) 稀缺性在 RLUK 集体馆藏中很常见，在 RLUK 成员馆的馆藏中鲜有重复，增加了集体馆藏的广度和深度；

(4) RLUK 集体馆藏中的印刷图书有将近 46 万个不同的主题；

(5) RLUK 集体馆藏相似于，同时也不同于研究图书馆协会(ARL)的集体馆藏：例如，相当大比例(42%)的 RLUK 印刷图书与 ARL 馆藏相重复，但是更大比例(58%)的印刷图书是不同的。

(编译自：http://www.oclc.org/en-US/news/releases/2016/201601sheffield.html)

(本刊讯)